Memory Waveform Plots

FTPM additions

Robert Goodwin

Thu, Nov 18, 2010

This note describes changes made to the local application `FTPM`, which supports the Acnet `FTPMAN` protocols. The purpose is to allow snapshot plotting of memory data without reference to specific digitizer hardware. Its need is prompted by the new Booster BLM digitizer hardware, designed by Craig Drennan, for which, after another LA reads out the data from the digitizer hardware FIFO, the 80 μs samples of beam loss are converted to an array of 16-bit data words proportional to the logs of the integrated loss. These data words are to be plotted by the new scheme.

A new `CINFO` system table entry type# 6 format is defined to assist with this support:

Field	*Size*	*Meaning*
`infoSz`	1	size of entry, either 16 or 24 bytes
`infoTy`	1	entry type# = 6
`infoCh`	2	analog Chan#
`nPts`	2	#points in data array
`spar2`	2	(n.u.)
`baseAddr`	4	memory base address of data array
`digRate`	4	data rate in Hz
`digEvt`	2	(opt.) clock event# starting delay for filling data array, say, `0x0010`
`digDly`	2	(opt.) delay after clock event, in μs
`spar4`	4	(n.u.)

In the case that `infoSz` is 16 bytes, it is assumed that the data array was started at Booster reset event time without delay. The `infoCh` field, found in any `CINFO` entry, specifies a channel# as found in an Acnet device reading property's `SSDN` sent to the front end in the snapshot request. The `baseAddr` specifies the fixed base of the data array to be sampled. The data rate corresponding to the data array elements is `digRate`.

The assumption for this support is that there is an array of data available for sampling on every operating cycle of the front end, either 10 Hz or 15 Hz. Soon after Micro-P Start time, when the front end begins its normal cyclic work, including updating the local data pool and fulfilling active data requests for which replies are due on the current cycle, the data buffer is assumed complete and available. A snapshot request may be made with a sampling rate and a number of points such that multiple cycles are encompassed. In this case, there may be data points that are unavailable to be included in the plot, but the returned data must adhere to the sampling rate time line; therefore, some zeros are supplied that correspond to the missing data points.

Specific example

As a specific example, the new BLM data amounts to 500 points with a digitize rate of 12.5 KHz. Since this covers a span of 40 ms of time, and the cycle lasts 66 ms, there are 26 ms of missing data. (In these front ends, Micro-P Start must be at least 40 ms from Booster reset time.) This new support samples the 12.5 KHz data points as best it can to match the requested sampling rate, and it includes zero data for the end portions of each cycle. A request for 1000 points sampled at 1 KHz will encompass one second of time. About 40 of the 500 points are sampled each cycle, beginning at the first point, followed by 26 zeros. When the entire snapshot data is plotted, it is hoped that the presence of these zeros will be obvious.

The snapshot data request may specify a reference event and a delay, so the first data point to be sampled occurs at that event plus delay. In addition, the data buffer may optionally be associated with an event plus delay via the optional `CINFO` entry fields above. The logic of this support awaits the reference event plus delay, and it begins sampling waveform data on the first cycle for which data is available that can fulfill the request. (On the first cycle, the first sampled data point may not be from the beginning of the data buffer.)

To flesh out the sampling logic a bit more, on each cycle for which data is available to satisfy the request, the data buffer is sampled from its start in "suitable steps" according to the requested sample rate. (A suitable step may not be an integral number of points. For example, it may be every 3 or 4 points, or it may be every 12 or 13 points; it depends on the ratio of the digitizer rate to the sampling rate.) Until a point in the data buffer is reached that corresponds to the time of the reference event plus delay, no sampled points are copied into the reply buffer. Copying continues until a suitable step passes the last point in the data buffer, or the reply buffer is full. The point in the reply buffer where copying left off is used as a reference for where to continue the copying of sampled data on the next cycle. Note that it is assumed that the data buffer is only available for one cycle; it is overwritten on the following cycle with new data.

Implementation details

Although `FTPM` is the largest (24K) local application, it does have some modularity in its organization that helps in adding new generic memory waveform support. Code was added to the `RqTiming1` function to answer the "what support do you have for this `SSDN`?" request, and related support was added to `RqSnapshot7` to recognize this new case. Two new functions were written. `SnapMem` is called by `RqSnapshot7` to help initialize the request, and `MemMon` is called by `UpdSnapshot`, which is itself called to monitor a snapshot request when active data requests (of all stripes) are being fulfilled shortly after the local data pool is refreshed each cycle.

Reviewing the nature of snapshot support in a front end, a request is received from a client, and related support structures are initialized by `FTPM` to support the request while it is active. The main structure is a "request block" that is inserted into the linked list of active requests. A related structure is the reply message block. For snapshot requests, the front end returns reply messages, say 2-3 times a second, to update the requester on the progress in collecting the requested snapshot data. When the internal reply buffer has been filled, a prompt status reply is sent to the client announcing that all data has been collected. The client can then send one or more one-shot requests to ask for the data that is being held. The front end keeps the data available until the request is canceled, or until the client sends a special message to ask that the same measurement be repeated.

Since the new generic memory waveform support is independent of any specific hardware interface, it is hoped it may be useful in other projects. It allows plotting across multiple cycles, but it is not a requirement. A suitably formed request could ask for a snapshot of an array of data that is fulfilled in a single cycle.